WHAT IS CLAIMED IS:

1. A method comprising:

converting memory access instructions in a source code into a standard format;

generating partitions containing formatted memory access instructions;

generating a match set, the match set including matches of instruction patterns to the formatted memory access instructions in the partitions; and

transforming the matches to vector memory access instructions.

2. The method of claim 1 in which converting comprises converting memory access instructions that read or write less than a minimum data access unit (MDAU) to memory access instructions that read or write a multiple of the minimum data access unit.

3. The method of claim 2 in which converting further comprises transforming the memory access instructions that read or write the multiple of the minimum data access unit to a format including a base address plus an offset.

4. The method of claim 1 in which generating partitions comprises:

generating a data flow graph containing basic blocks including the memory access instructions; and

for each basic block, applying a set of rules.

5. The method of claim 4 in which applying comprises limiting a subnode of a partition to memory access instructions directed to a specific memory bank.

6. The method of claim 5 in which applying further comprises limiting the subnode of a partition to a memory read or a memory write.

7. The method of claim 5 in which the memory bank is a static random access memory (SRAM).

8. The method of claim 5 in which the memory bank is a dynamic random access memory (DRAM).

9. The method of claim 5 in which the memory bank is a scratchpad memory.

10. The method of claim 5 in which the memory back is an EEPROM.

11. The method of claim 5 in which the memory back is flash memory.

12. The method of claim 5 in which the memory back is a NVRAM.

13. The method of claim 1 in which the instruction patterns comprise a pattern describing instruction semantics.

14. The method of claim 1 in which the vector memory access instructions comprise single memory access instructions representing multiple memory accesses to a type of memory.

15. A compilation method comprising:

converting memory access instructions that read or write less than a minimum data access unit (MDAU) to memory access instructions that read or write a multiple of the minimum data access unit;

converting the memory access instructions into a format including a base address plus an offset;

grouping subsets of the converted memory access instructions into partitions; and

vectorizing the converted memory access instructions in the subsets that match instruction patterns.

16. The compilation method of claim 15 in which grouping comprises:

generating a data flow graph containing basic blocks including memory access instructions; and

generating subnodes in partitions, the subnodes including memory access instructions directed to a memory bank and performing the same operation.

17. The compilation method of claim 16 in which the operation is a read.

18. The compilation method of claim 16 in which the operation is a write.

19. The compilation method of claim 16 in which the memory bank is a static random access memory (SRAM).

20. The compilation method of claim 16 in which the memory bank is a dynamic random access memory (DRAM).

21. The compilation method of claim 16 in which the memory bank is a scratchpad memory.

22. The compilation method of claim 16 in which the memory bank is an EEPROM.

23. The compilation method of claim 16 in which the memory bank is flash memory.

24. The compilation method of claim 16 in which the memory bank is NVRAM.

25. The compilation method of claim 15 in which the instruction patterns comprises instruction semantics.

26. The compilation method of claim 25 in which the instruction semantics comprises segments.

27. A computer program product tangibly embodied in an information carrier, for vectorizing memory access instructions, the computer program product being operable to cause data processing apparatus to:

convert memory access instructions residing in a source code into a standard format;

generate partitions containing formatted memory access instructions;

generate a match set, the match set including matches of instruction patterns to the formatted memory access instructions in the subsets; and

transform the matches to vector memory access instructions.

28. The product of claim 27 in which converting comprises converting memory access instructions that read or write less than a minimum data access unit to memory access instructions that read or write a multiple of the minimum data access unit.

29. The product of claim 28 in which converting further comprises transforming the memory access instructions that read or write the multiple of the minimum data access unit to a format including a base address plus an offset.

30. The product of claim 27 in which generating partitions comprises:

generating a data flow graph containing basic blocks including memory access instructions; and generating subnodes in partitions, the subnodes including memory access instructions directed to a memory bank and performing the same operation.